Process Mining with Common Sense

Diego Calvanese[1,2], Sanja Lukumbuzya[3],
Marco Montali[1], and Mantas Simkus[3]

[1] Free University of Bozen-Bolzano, Bolzano (Italy)
{calvanese,montali}@inf.unibz.it
[2] Umeå University, Umeå (Sweden)
[3] Vienna University of Technology, Vienna (Austria)
sanja.pavlovic@tuwien.ac.at, simkus@dbai.tuwien.ac.at

Abstract. We argue that, with the growth of process mining in breadth (variety of covered tasks) and depth (sophistication of the considered process models), event logs need to be augmented by commonsense knowledge to provide a better input for process mining algorithms. This is crucial to infer key facts that are not explicitly recorded in the logs, but are necessary in a variety of tasks, such as understanding the event data, assessing their compliance and quality, identifying outliers and clusters, computing statistics, and discovering decisions, ultimately empowering process mining as a whole.

Keywords: Commonsense knowledge · process mining · event data

1 Description of the problem

Process mining is experiencing a golden age, with a flourishing academic community and an increasing adoption by industry. The field evolved from the seminal works on control-flow process discovery from the late 90s to a full-fledged arsenal of techniques for process improvement. Contemporary process mining techniques indeed range from advanced forms of process discovery to sophisticated forms of process enhancement, conformance checking, and online operational support. At the same time, such techniques have been refined and extended over the years to target increasingly sophisticated process models. At the frontier of research, we indeed find models that tackle multiple perspective at once (such as control-flow, data, and decisions - see, e.g., [16, 5, 17, 10, 13]), and that are moving their focus from single cases to multiple, co-evolving objects [1, 12, 4, 18, 14].

Mirroring this trend, the event data constituting the input fuel for process mining algorithms evolved from simple (multi)sets of traces containing sequences of propositional events to richer traces annotated with different types of data attributes [16, 6, 15], finally culminating in full-fledged networks of events related to each other through multiple objects and relationships [2, 11]. This evolution is also witnessed by the employed data formats from event logs, moving from the well-established XES (`https://xes-standard.org`) standard to recent proposals such as XOC [15] and OCEL (`http://ocel-standard.org`).

In this complex spectrum, it is common practice to make the assumption that the event log used as input for a process mining project explicitly contains enough relevant facts to faithfully apply the intended algorithms. We argue that, even when the recorded event data are of high quality, this is in general a too strong assumption. Going a step further, a human, or, to be more precise, a domain expert, would be able to easily reconstruct all the relevant facts from the log *by applying their own commonsense knowledge*.[4] The following two examples ground this observation in two relevant settings: reconstruction of the state of affairs in a process execution, and identification of outlier vs impossible traces.

Example 1. We focus on a typical object-centric process dealing with an order-to-cash scenario where multiple orders may be handled together, possibly transferring order lines from one to the other, and applying coupons for getting a discount. We assume that there is a database listing the different available item types, and indicating, for each item type, the corresponding unit cost. Consider, in this context, the following sequence of events (working on two, related orders, where we omit the responsible persons for simplicity)[5]:

EVENT	TIMESTAMP	ORDER	ITEM TYPE	ATTRS
create order	01/06/2021 9:00	o_1		
add item	01/06/2021 9:01	o_1	laptop	
create order	01/06/2021 9:03	o_2		
add item	01/06/2021 9:04	o_1	mouse	
add item	01/06/2021 9:08	o_2	laptop	
transfer content	01/06/2021 9:15	o_2, o_1		
insert coupon	01/06/2021 9:20	o_1		20%
insert coupon	01/06/2021 9:21	o_1		20%
pay	01/06/2021 9:25	o_1		

By looking into this log and the price table mentioned above, a domain expert may easily reconstruct the final state of each order, including the overall amount associated to order o_1 that is paid when executing the last entry. This is, in turn, relevant for a number of process mining tasks, such as classification of orders and clustering of their corresponding traces, or mining decisions to understand which rules are applied by customers to choose under which conditions an order is eventually paid or not.

In the specific sample log, what happens is that two orders are created concurrently, only later on realising that they can be merged into a single order, which is then finally paid. Specifically, the content of order o_2 is transferred into order o_1, which ultimately contains two laptops and one mouse, and is paid after applying a discount.

[4] We use "commmonsense knowledge" as an umbrella term for general knowledge about the world, as well as general knowledge about a specific organisational setting.

[5] We use a tabular format that resembles the input format used in [2] for discovery.

How is it possible to reconstruct these important, implicit facts, which are not explicitly contained in the (sound and complete) given sequence of events? *Thanks to the application of commonsense knowledge (both general and domain-specific).* In fact, commonsense knowledge is used to understand that:

- when adding an item to an order, the other, previously added items will continue to be included in that order;
- when the content of order o_2 gets transferred into another order o_1, the effect is that o_2 becomes empty, while o_1 contains all and only those items that were contained in o_1 or o_2.

While these two examples relate to general, commonsense knowledge, there is also domain-specific commonsense knowledge that only domain experts have. Two key pieces of domain-specific knowledge in our example would be needed to answer the following questions:

- What is the resulting state of an order after its content is being transferred to another order? Does it stay active as an empty order? Or is it implicitly cancelled?
- Is it possible to apply multiple discount coupons to the same order? If so, are the discount percentages applied in cascade, or always to the overall amount of the order?

Notice that reconstructing the state of affairs and answering the questions above *are orthogonal aspects to the quality of event data, as they concern implicit facts related to the cumulative effect of events, not the explicit events themselves.*

The next example is inspired by [3], though there they focus on understanding process modeling constructs, whereas here we keep our focus on event data.

Example 2. Consider a simplified version of a case-centric logistic process where each process instance focuses on the prepare- of a single package. We do not have data attributes, just sequences of events (with their frequency). Consider in particular the following log, containing three process variants:

Variant 1 (970 traces)	**Variant 2** (20 traces)	**Variant 3** (10 traces)
receive payment		
prepare package	prepare package	prepare package
load package	load package	
deliver package	deliver package	deliver package
	receive payment	receive payment

The most common Variant 1 represents the standard execution of the process: after receiving the payment from the customer, the package is prepared, loaded in a truck, and delivered to the customer. Variant 2 represents an outlier behaviour, where a package is prepared and delivered without a prior payment, which is in fact received only after the delivery is completed. Interestingly, a domain expert could judge that this behaviour is an outlier behaviour event without looking at the frequency, if it is common practice in the company to only deliver material that has been paid before. This shows the multitude of usages of commonsense knowledge.

What discussed for Variant 2 is taken to the extreme when looking at Variant 3. There, a package is prepared without a prior payment. What appears even stranger, though, is that the package is not loaded in the truck, but is nevertheless delivered, finally resulting in a payment. This appears strange because, again due to commonsense knowledge, a human would immediately understand that a package cannot be delivered in a truck if it has not been loaded there upfront. This, in turn, would immediately lead to label Variant 3 as a variant that cannot be simply judged as an outlier, but requires instead further inspection to understand whether it relates to: incomplete/faulty logging of activities (e.g., the package was actually loaded, but this operation has not been logged), presence of a fraud (money received without any material delivery), or other. Understanding what is happening with Variant 3 is in turn crucial towards compliance assessment, classification of outliers, computation of statistics, data quality, etc.

The main issue is that this commonsense knowledge is not explicitly communicated to process mining algorithms: it either stays in the head of domain experts, and only indirectly emerges in how such experts interact with such algorithms and the results they produce, or it gets embedded into ad-hoc, hardly interpretable pieces of code used to (pre-)process event data. At the same time, eliciting such knowledge is a notoriously difficult problem.

All in all, the problem is then: *How can we suitably augment event data with commonsense knowledge, to improve the faithfulness and quality of process mining without introducing too much additional modelling effort?*

2 Why is this an important problem

This problem relates to the grand challenge of *garbage-in garbage-out* in process mining and, more in general, in data science. While the community has extensively worked on data quality targeting the data explicitly contained in an event log, no attention has been devoted to the insights that can be obtained through commonsense reasoning on such data. The two examples above indicate that even in very simple examples this is of utmost importance to support, and better inform, process mining algorithms.

At the same time, the elicitation and usage of commonsense knowledge is one of the central open problems in (general) artificial intelligence [8]. Focusing on a concrete setting, such as that of process mining, grounds this problem in a concrete context, paving the way to more accessible results that could in turn provide insights on how to advance with the problem in its full generality.

3 Relation with existing work

Within the BPM community, this problem is largely unexplored. The literature has so far mainly targeted the problem of integrating structural knowledge with process models (see, e.g., [4]), or in providing richer ontological foundations for

process models themselves (see, e.g., [3]). The huge body of work on data quality for event data is complementary, and in synergy, with the problem described here. Progress within artificial intelligence on commonsense knowledge and commonsense reasoning, in terms of general inference mechanisms [7] and knowledge bases targeting specific domains that are relevant for BPM [9], provides a solid basis to attack the problem presented here.

4 Initial ideas towards solving the problem

As pointed out in [8], commonsense reasoning is usually tackled using either knowledge-based or machine learning-based approaches, with very limited interactions between them. Also crowd-sourcing approaches exist. In the setting considered here, we advocate the knowledge-based approach, which is particularly effective when dealing with qualitative reasoning [8], of central importance when applying common sense to event data.

Specifically, we are investigating the adoption and further development of techniques in the area of Knowledge Representation & Reasoning (KR&R). KR&R is developing methods for capturing human knowledge in machine-processable knowledge bases, which can used by automated reasoning engines to provide non-trivial insights to the users. If knowledge bases consist of raw data enriched with common sense knowledge, then reasoning engines can be used to infer useful new facts that follow logically from the data and the captured human knowledge. This can for example be exploited to alleviate information incompleteness and to identify inconsistencies in the data. KR&R already offers several techniques related to the challenges discussed above, such as reasoning about actions and change, non-monotonic reasoning, rule-based languages, and description logics. These individual techniques have different strengths, which need to be combined in order to address the multifaceted needs discussed above.

As a first step, we are developing a new lightweight formalism that combines structural knowledge (to represent the relevant classes and relationships within an organisation), temporal/dynamic operators (to represent the process dynamics), and nonmonotonic features (essential to capture what does *not* change when an event occurs), while guaranteeing efficient inference mechanisms. To the best of our knowledge, a formalism balancing all these different ingredients is missing.

The second step will be to understand how minimal knowledge bases expressed in this formalism can be elicited and (re)used to augment event data, analysing the impact on applicability and quality of process mining techniques.

Acknowledgements. This research has been partially supportedby the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation, by the Italian Basic Research (PRIN) project HOPE, by the EU H2020 project INODE, grant agreement 863410, by the CHIST-ERA project PACMEL, by the project IDEE (FESR1133) funded by the European Regional Development Fund (ERDF) Investment for Growth and Jobs Programme 2014-2020, and by the Free University of Bozen-Bolzano through the projects KGID, GeoVKG, STyLoLa, and VERBA.

References

1. van der Aalst, W.M.P.: Object-centric process mining: Dealing with divergence and convergence in event data. In: Proc. of SEFM. LNCS, vol. 11724, pp. 3–25. Springer (2019)
2. van der Aalst, W.M.P., Berti, A.: Discovering object-centric Petri Nets. Fundamenta Informaticae **175**(1-4), 1–40 (2020)
3. Adamo, G., Di Francescomarino, C., Ghidini, C., Maggi, F.M.: Beyond arrows in process models: A user study on activity dependences and their rationales. Information Systems **100**, 101762 (2021)
4. Artale, A., Kovtunova, A., Montali, M., van der Aalst, W.M.P.: Modeling and reasoning over declarative data-aware processes with object-centric behavioral constraints. In: Proc. of BPM. LNCS, vol. 11675, pp. 139–156. Springer (2019)
5. Burattin, A., Maggi, F.M., Sperduti, A.: Conformance checking based on multi-perspective declarative process models. Expert Systems with Applications **65**, 194–211 (2016)
6. Calvanese, D., Kalayci, T.E., Montali, M., Tinella, S.: Ontology-based data access for extracting event logs from legacy data: The onprom tool and methodology. In: Proc. of BIS. LNBIP, vol. 288, pp. 220–236. Springer (2017)
7. Davis, E.: Logical formalizations of commonsense reasoning: A survey. J. of Artificial Intelligence Research **59**, 651–723 (2017)
8. Davis, E., Marcus, G.: Commonsense reasoning and commonsense knowledge in artificial intelligence. Communications of the ACM **58**(9), 92–103 (2015)
9. Davis, E., Marcus, G., Frazier-Logue, N.: Commonsense reasoning about containers using radically incomplete information. Artificial Intelligence **248**, 46–84 (2017)
10. De Smedt, J., Hasic, F., vanden Broucke, S.K.L.M., Vanthienen, J.: Holistic discovery of decision models from process execution data. Knowledge Based Systems **183** (2019)
11. Esser, S., Fahland, D.: Multi-dimensional event data in graph databases. CoRR Technical Report arXiv:2005.14552, arXiv.org e-Print archive (2020), `https://arxiv.org/abs/2005.14552`
12. Fahland, D.: Describing behavior of processes with many-to-many interactions. In: Proc. of Petri Nets. LNCS, vol. 11522, pp. 3–24. Springer (2019)
13. Felli, P., Gianola, A., Montali, M., Rivkin, A., Winkler, S.: CoCoMoT: Conformance checking of multi-perspective processes via SMT. In: Proc. of BPM. LNCS, vol. 11157, pp. 219–235. Springer (2021)
14. Ghilardi, S., Gianola, A., Montali, M., Rivkin, A.: Petri nets with parameterised data - modelling and verification. In: Proc. of BPM. LNCS, vol. 12168, pp. 55–74. Springer (2020)
15. Li, G., de Murillas, E.G.L., Medeiros de Carvalho, R., van der Aalst, W.M.P.: Extracting object-centric event logs to support process mining on databases. In: Proc. of CAiSE Forum. LNBIP, vol. 317, pp. 182–199. Springer (2018)
16. Lu, X., Nagelkerke, M., van de Wiel, D., Fahland, D.: Discovering interacting artifacts from ERP systems. IEEE Trans. Serv. Comput. **8**(6), 861–873 (2015)
17. Mannhardt, F., de Leoni, M., Reijers, H.A., van der Aalst, W.M.P.: Data-driven process discovery - revealing conditional infrequent behavior from event logs. In: Proc. of CAiSE. LNCS, vol. 10253, pp. 545–560. Springer (2017)
18. Polyvyanyy, A., van der Werf, J.M.E.M., Overbeek, S., Brouwers, R.: Information systems modeling: Language, verification, and tool support. In: Proc. of CAiSE. LNCS, vol. 11483, pp. 194–212. Springer (2019)